**3.**

class Car:

# Class variable

wheels = 4

def \_\_init\_\_(self, make, model, year):

# Instance variables

self.make = make

self.model = model

self.year = year

def description(self):

print(f"{self.year} {self.make} {self.model} with {Car.wheels} wheels.")

car1 = Car("Honda", "Civic", 2022)

car2 = Car("Toyota", "Corolla", 2021)

print(car1.make)

print(car2.year)

**OUTPUT:**



**4.**

class Student:

def \_\_init\_\_(self):

self.percentage = None

self.chemistry\_marks = None

self.cs\_marks = None

self.physics\_marks = None

self.math\_marks = None

self.name = None

self.roll\_no = None

self.it\_marks = None

def getstudentdetails(self):

self.name = input("Enter name: ")

self.roll\_no = input("Enter roll no: ")

self.math\_marks = int(input("Enter Math marks: "))

self.physics\_marks = int(input("Enter Physics marks: "))

self.cs\_marks = int(input("Enter C.S marks: "))

self.it\_marks = int(input("Enter I.T marks: "))

self.chemistry\_marks = int(input("Enter Chemistry marks: "))

def printresult(self):

self.percentage = int(

(self.math\_marks + self.physics\_marks + self.cs\_marks + self.it\_marks + self.chemistry\_marks) / 500 \* 100)

print("Name: ",self.name, "\nRoll No: ", self.roll\_no, "\nPercentage:", self.percentage)

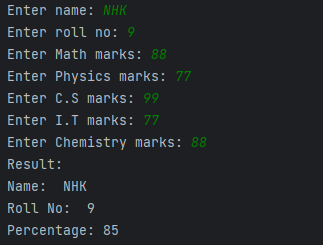
s1 = Student()

s1.getstudentdetails()

print("Result: ")

s1.printresult()

**OUTPUT:**



**5.**

class Bank:

bankbalance = 0

def account(self):

self.account\_name = input("Enter name: ")

self.age = input("Enter age: ")

self.amount = int(input("Enter amount: "))

Bank.bankbalance = Bank.bankbalance + self.amount

@classmethod

def showbankbal(cls):

print("Total Balance: ", cls.bankbalance)

LC = dict()

while True:

newAccount = Bank()

key = newAccount.account()

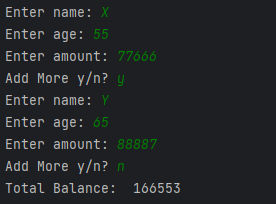
LC.setdefault(key, newAccount)

ch = input("Add More y/n? ")

if ch == "n": break

Bank.showbankbal()

**OUTPUT:**

****

**6.**

class Person:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

def get\_age(self):

return self.\_age

def set\_age(self, age):

if age < 0 or age > 120:

raise ValueError("Invalid age")

self.\_age = age

age = property(get\_age, set\_age)

person = Person("John", 25)

# Get the age

print(person.age)

# Set the age

person.age = 30

print(person.age)

person.age = -5

**OUTPUT:**

